



Module: Systèmes Embarqués

Chapitre 4: LES MICROCONTRÔLEURS

El Kefi HLEL

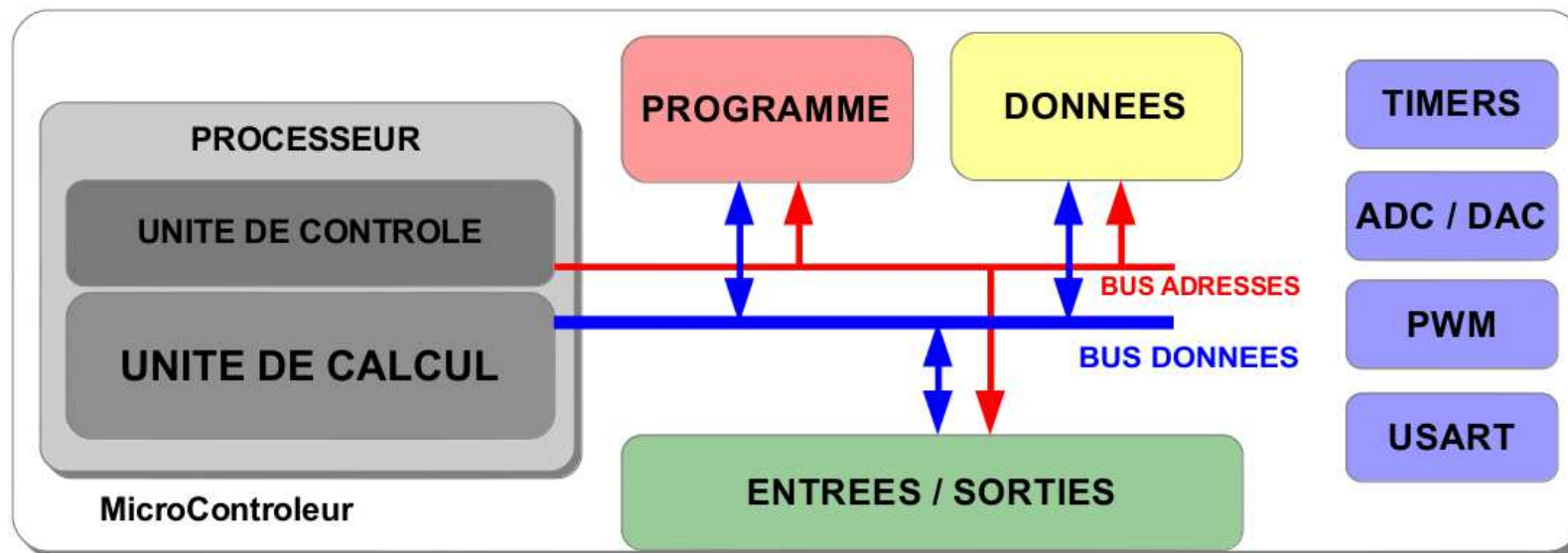
Février 2014

PRÉSENTATION DES MICROCONTRÔLEURS

ARCHITECTURE INTERNE

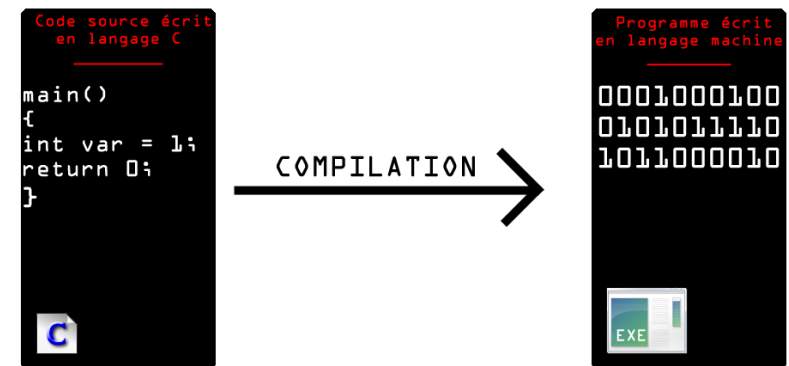
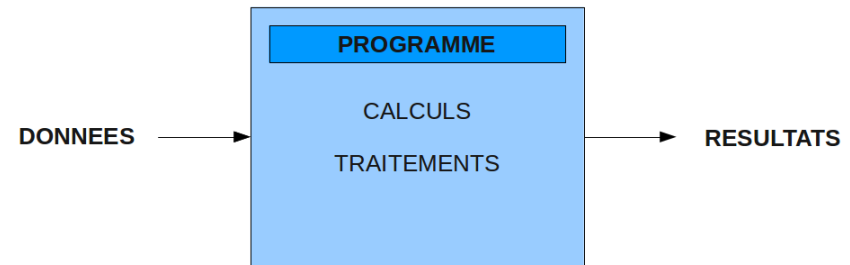
Un microcontrôleur intègre sur un unique *die* (circuit intégré) :

- Un processeur (CPU), avec une largeur du chemin de données allant de 4 bits pour les modèles les plus basiques à 32 ou 64 bits pour les modèles les plus évolués
- De la mémoire vive (RAM) pour stocker les données et les variables ;
- De la mémoire morte (ROM) pour stocker le programme (EPROM, EEPROM, mémoire flash, ...)
- Une horloge pour le cadencement. Il peut être réalisé avec un quartz, un circuit RC ou une PLL
- Des périphériques, capables d'effectuer des tâches spécifiques

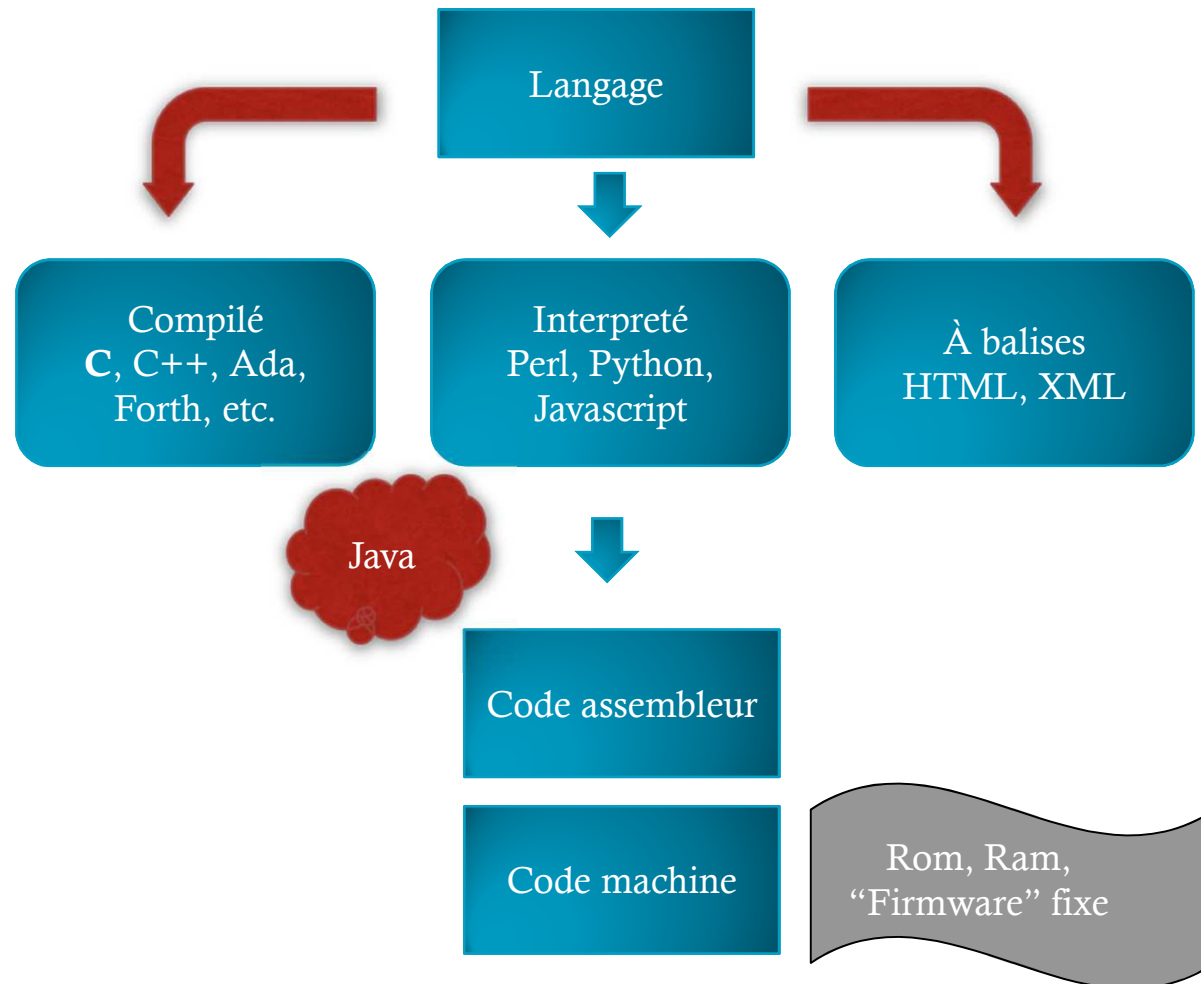


LANGAGE DE PROGRAMMATION

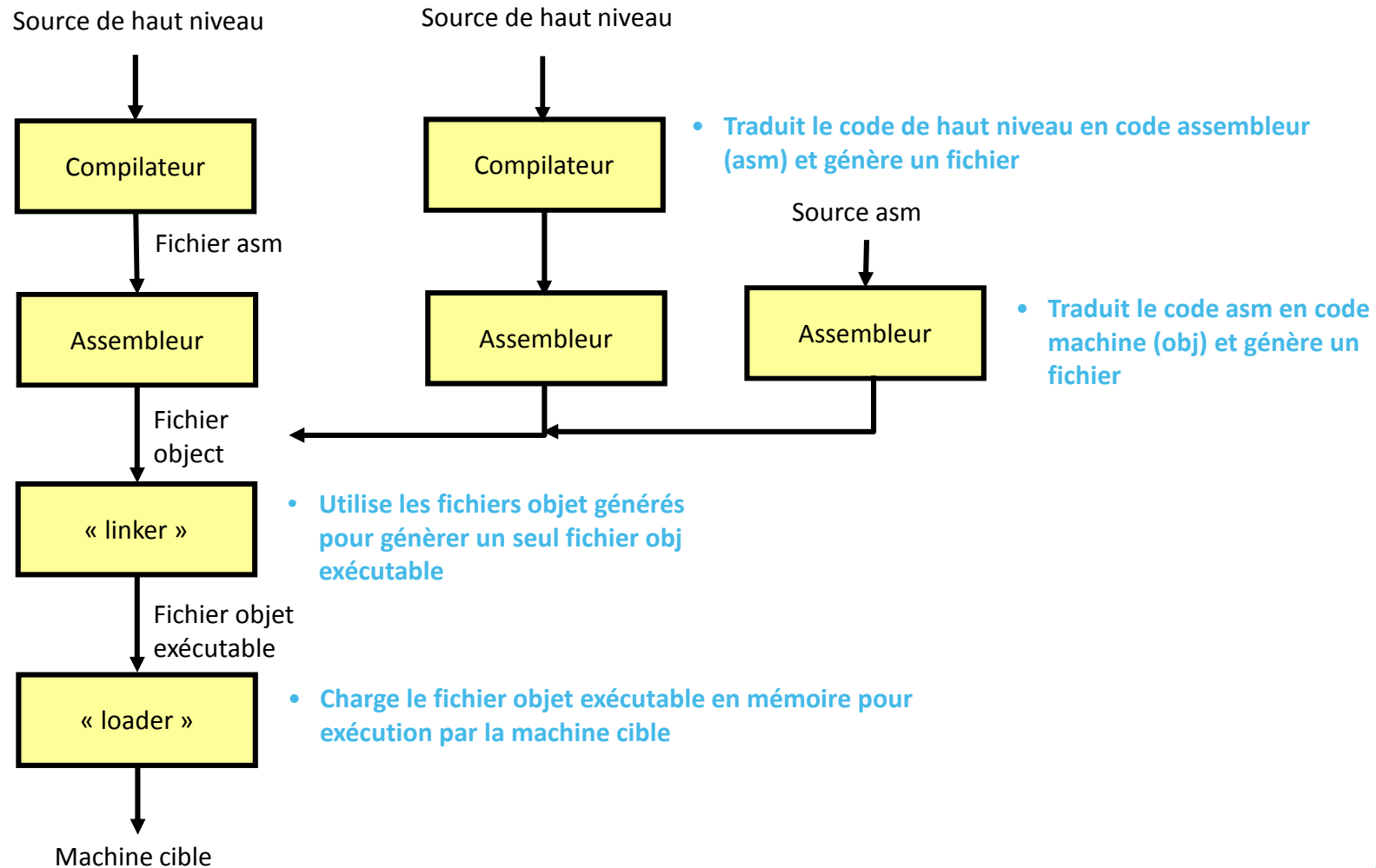
- Rôle d'un programme
 - Lire des données en entrées (capteur)
 - Effectuer des calculs sur ces données
 - Fournir un résultat en sortie (actionneur)
- Langage de haut niveau
 - Notion d'abstraction
 - Proche du langage humain
 - Exemples de langages: C, Java, C++, python, ...
- Langage machine
 - Langage natif d'un processeur. Il est composé d'instructions et de données à traiter codées en binaire.



LANGAGES DE HAUT NIVEAU



CYCLE DE GÉNÉRATION DE PROGRAMMES EN LANGAGE DE HAUT NIVEAU



ENVIRONNEMENTS DE PROGRAMMATION (1)

- À l'origine, les microcontrôleurs se programmaient en assembleur (langage bas niveau). L'assembleur pose toujours des problèmes pour la maintenance et l'évolution des logiciels embarqués
- On utilise de plus en plus des langages de haut niveau (ex le langage C) qui est capable de faciliter la programmation de microcontrôleurs toujours plus puissants
- Les compilateurs C présentent généralement certaines restrictions liées aux spécificités des microcontrôleurs (par exemple mémoire limitée)
- Avec l'augmentation de la puissance et de la quantité de mémoire de stockage (FLASH) disponible dans les microcontrôleurs, les programmes de ces derniers peuvent désormais être écrits en C++

ENVIRONNEMENTS DE PROGRAMMATION (2)

- Des simulateurs sont disponibles pour certains microcontrôleurs. Les développeurs peuvent ainsi analyser le comportement du microcontrôleur et du programme, comme s'il s'agissait du composant réel
- Un simulateur montre l'état interne du processeur, ainsi que celui de ses sorties. On peut de cette façon créer des conditions similaires à des implémentations physiques. Cela facilite donc l'analyse et la résolution des problèmes
- Certains logiciels de CAO simulent le comportement du microcontrôleur et des composants qui l'entourent. On peut ainsi afficher les états des entrées et sorties des différents composants au cours de l'exécution d'un programme

FAMILLES DE MICROCONTRÔLEURS

- La famille Atmel AT91
- La famille Hitachi H8
- La famille Intel 8051, 8085
- Les familles Freescale 68HC08, 68HC12
- La famille des PIC de Microchip
- La famille des ST6, ST7, STM8, ST10, STR7, STR9, STM32 de STMicroelectronics
- La famille MSP430 de Texas Instruments
- Les familles V800 et K0 de NEC
- etc ...

Exemple: Le circuit intégré d'Intel 8742, un microcontrôleur 8 bits qui comprend un processeur cadencé à 12 MHz, 128 octets de mémoire vive, une EPROM de 2048 bits, et des entrées-sorties sur une unique puce.



STMICROELECTRONICS

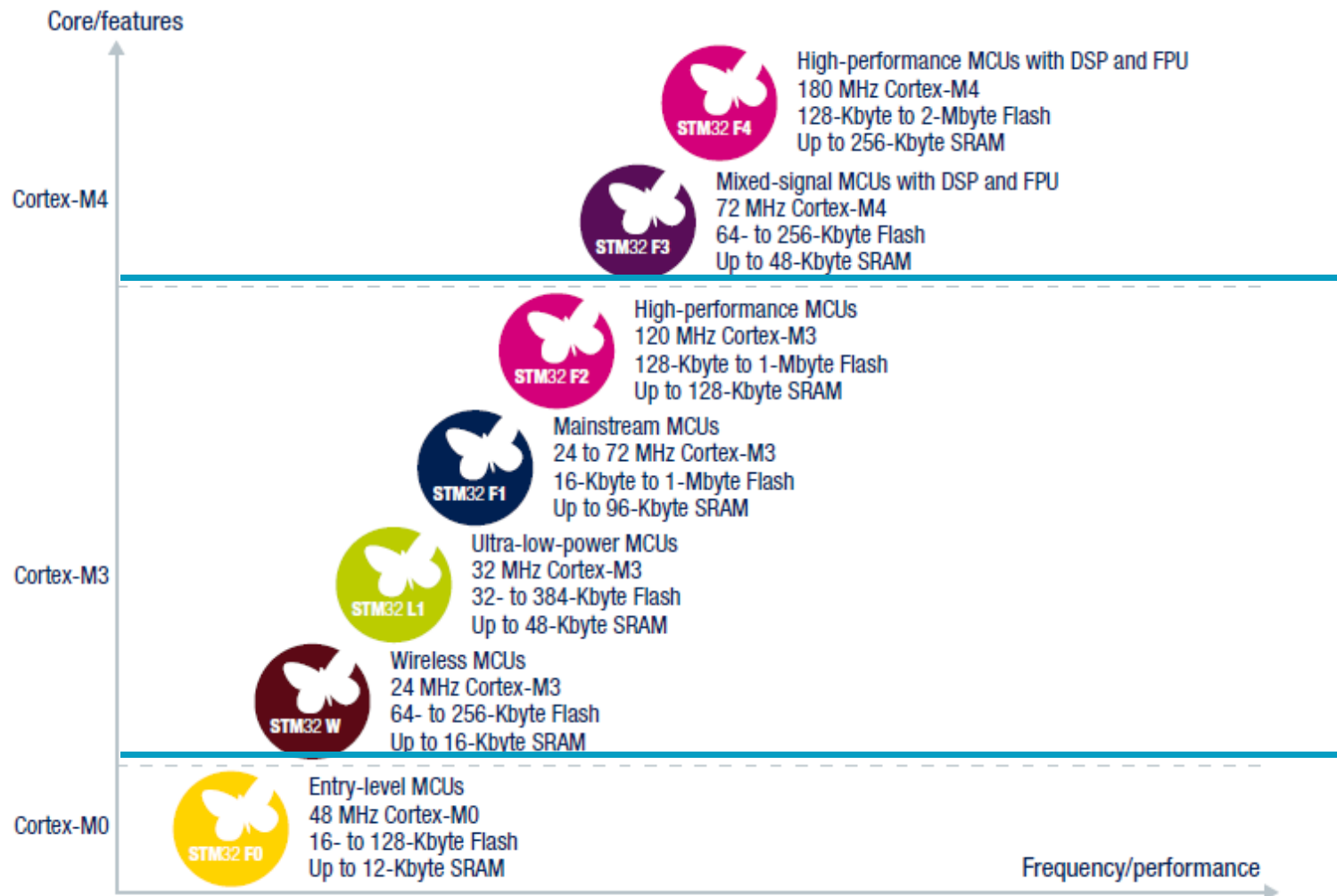
Société franco/italienne. Parmi ces activités on peut citer le développement de microcontrôleurs 8, 16 et 32 bits pour des applications industrielles



Samsung
GALAXY
Note III

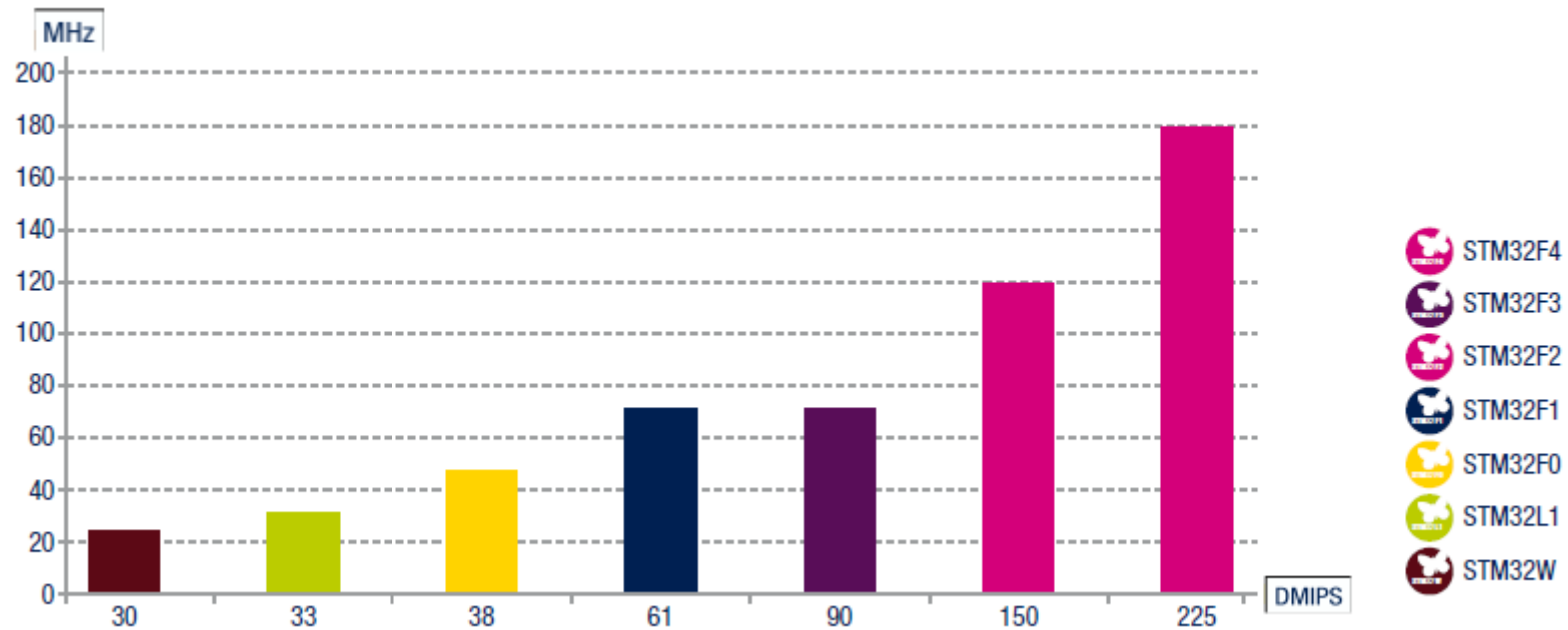


LA FAMILLE STM32



LES PERFORMANCES DES MICROCONTRÔLEURS STM32

STM32 PERFORMANCE



PRÉSENTATION DE STM32 F4



CARACTÉRISTIQUES DE STM32 F4

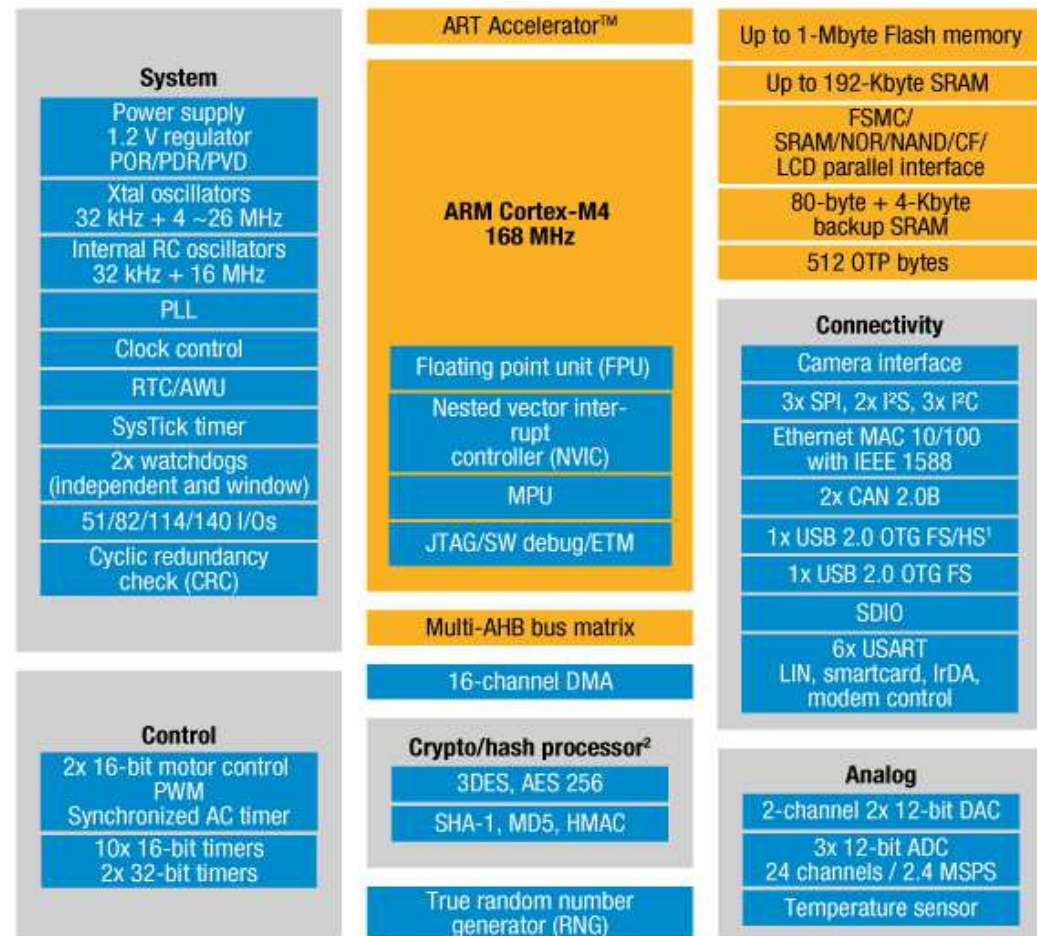
- Microcontrôleur ST basé sur le processeur ARM Cortex-M4
- Processeur RISC
- Architecture Harvard : bus instructions et bus données sont séparés
- Mode de gestion de la consommation
- Supporte uniquement le jeu d'instruction Thumb2

Key features

Real-time performance Cortex Intelligent Processors by ARM 	Outstanding power efficiency 	Superior and innovative peripherals 	Maximum integration 	Extensive tools and software 
Cortex-M4 with F_{CPU} 168 MHz/210 DMIPS	< 1 μA V_{BAT} RTC, Ultra low dynamic consumption, 1.7 to 3.6 V V_{DD}	Faster peripherals, 2 full duplex I²S, RTC with sub second accuracy...	1 MByte Flash, 192-Kbyte SRAM...	CMSIS DSP library, Matlab support, various IDE starter kits, RTOS, and stacks

BLOC DIAGRAMME DE STM32 F4

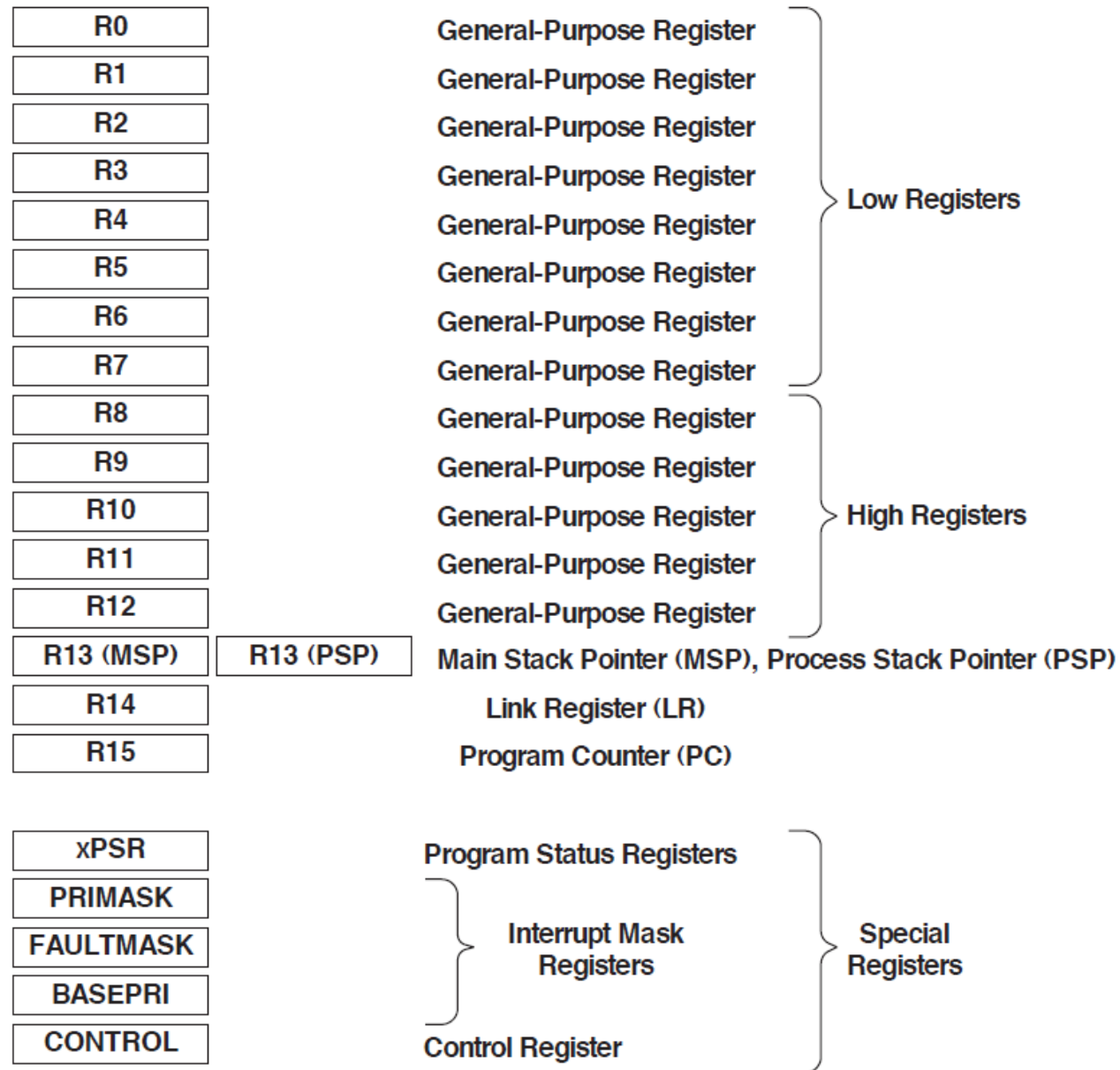
- Cœur : processeur 32 bits ARM Cortex™-M4 avec unité à virgule flottante (FPU)
- Mémoires
 - Jusqu'à 1 Mo de mémoire Flash
 - Jusqu'à 192+4 Ko de mémoire SRAM
- Interface parallèle LCD
- Gestion de l'horloge, de la réinitialisation et de l'alimentation
- Modes: attente, arrêt et veille
- 3 convertisseurs A/N 12 bits
- 2 convertisseurs N/A 12 bits
- Jusqu'à 17 temporisateurs
- Jusqu'à 140 ports d'E/S
- Jusqu'à 15 interfaces de communication
- Interface de caméra parallèle 8 à 14 bits jusqu'à 54 Mo/s



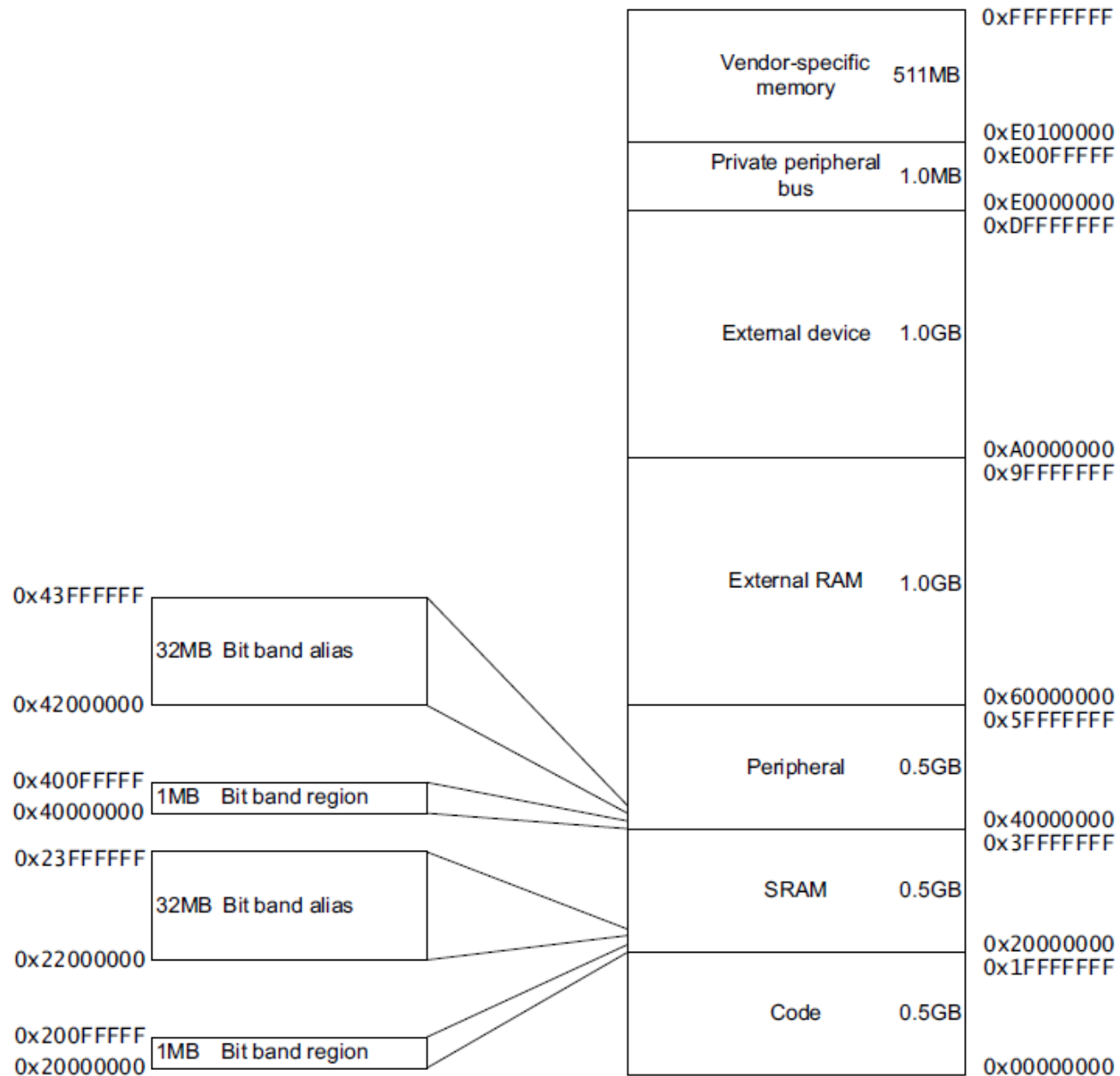
Notes:

1. HS requires an external PHY connected to the ULPI interface
2. Crypto/hash processor on STM32F417 and STM32F415

STM32 F4 : LES REGISTRES



CORTEX-M4: MAPPING DE LA MÉMOIRE



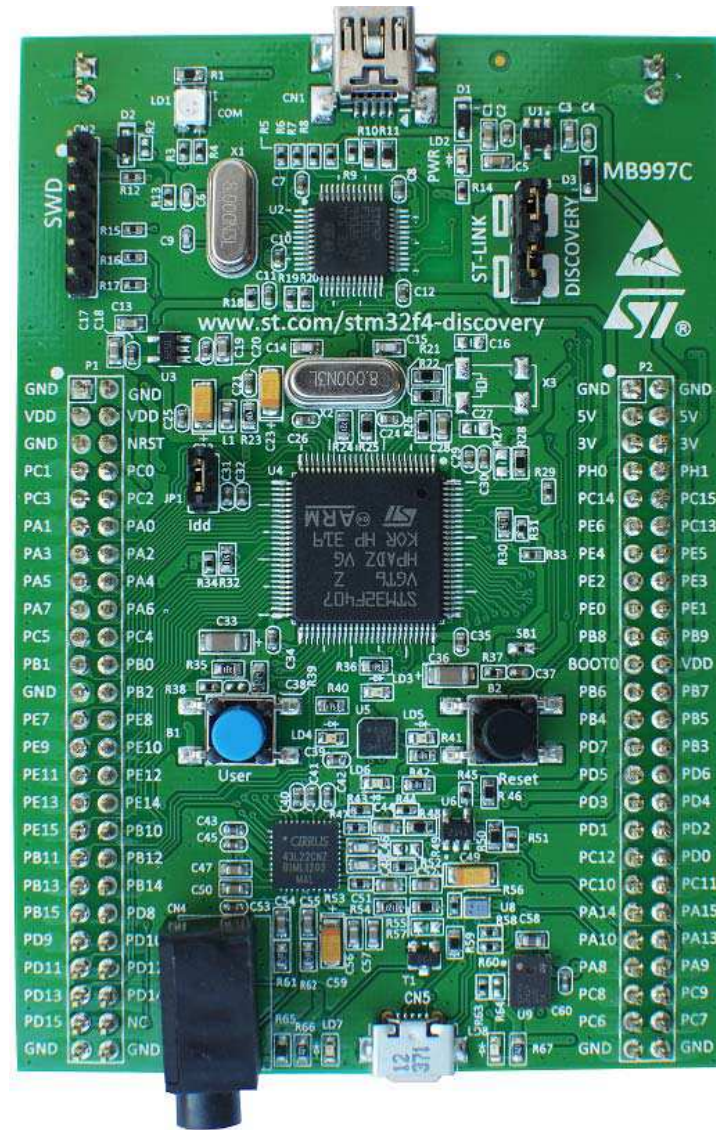
CARTE DE DÉVELOPPEMENT: STM32F4DISCOVERY



STM32F4DISCOVERY

DESCRIPTION DE LA CARTE (STM32F4DISCOVERY)

- Petite carte (7 cm x 10 cm) basée sur un cortex M4 modèle STM32F407VGT6, 1 MB de flash et 192 KB de RAM
- Elle est équipée en série d'un ST-Link (V2) permettant la programmation directe depuis un port USB, ce ST-Link pouvant également programmer tout autre microcontrôleur de la même famille
- Un détecteur de mouvement MEMS,
- Un micro numérique MEMS,
- Un convertisseur N/A audio avec amplificateur de haut-parleurs de classe D
- 4 voyants à LED
- 2 boutons-poussoirs (un bouton pour interagir avec le programme et un bouton « Reset »)
- Un connecteur micro USB type AB
- Peut être alimentée en 3, 5 volts ou par le port USB



ENVIRONNEMENT DE DÉVELOPPEMENT

- La carte de développement STM32F4DISCOVERY est une carte d'initiation à la programmation du STM32F4. Cette série de microcontrôleur contient un processeur Cortex-M4 qui peut exécuter des applications de traitement de signal numérique (filtrage, extraction de signaux ...) assez rapidement
- Pour pouvoir programmer sur la carte STM32F4, il nous faut un environnement de développement. L'outil de développement qui sera utilisé est « **Keil µVision** »
 - *Environnement de développement intégré*: un compilateur C/C++, débogueur. Combine la gestion de projet, l'édition, le débogage du code source du programme, et de la simulation complète dans un environnement puissant
 - *Facile à utiliser*: l'utilisateur peut commencer ses projets de développement en spécifiant le microcontrôleur qu'il veut utiliser
 - *Le débogueur* aussi peut être utilisé pour simuler les périphériques. De cette manière, l'utilisateur peut vérifier une application avant de télécharger le code sur son équipement système.

EXEMPLE D'APPLICATION: AUDIO PLAYBACK & RECORD

